

## IN THE CLAIMS

1. (Currently Amended) A system comprising:  
a memory storing a compressed image as a codestream in a first progression order;  
a progression order conversion parser to convert the codestream from the first  
progression order to a second progression order different than the first  
progression order by reading one or more markers of the codestream to  
determine a current type of progression, the one or more markers further  
indicating how data of the codestream should be handled during the  
progression order conversion, updating the one or more markers to specify a  
target type of progression, and outputting packets of the codestream in an order  
conforming to the second progression order indicated by the updated one or  
more markers,  
wherein the parser converts the codestream from the first progression order to an  
intermediate progression order and from the intermediate progression order to  
the second progression order, and  
wherein the intermediate progression order is a layer progression order and the second  
progression order is a target progression order other than the layer progression  
order.

2. (Cancelled)

3. (Currently Amended) The system defined in Claim ~~2~~1 wherein the intermediate progression order comprises a Layer-resolution-component-position progression of JPEG 2000.
4. (Original) The system defined in Claim 3 when the first progression order is one of the following group of JPEG 2000 progression orders: resolution-layer-component-position progression; resolution-position-component-layer progression; component-position-resolution-layer progression; and position-component-resolution-layer progression.
5. (Original) The system defined in Claim 1 wherein the parser:
  - determines where packets exist in the codestream based on at least one marker;
  - creates a structure specifying components in each packet; and
  - reorders packets in the codestream using the structure to map the first progression order to the second progression order.
6. (Original) The system defined in Claim 1 wherein the parser, in response to receiving a request, performs the conversion and sends the codestream in the second progression order.
7. (Previously Presented) The system defined in Claim 6 wherein the memory is part of a server that serves the image in response to requests, wherein the request is received in response to an activation by a client on a first image having the first progression order, and wherein in response to the request, a second image having the second progression order is presented to the client.

8. (Original) The system defined in Claim 7 wherein the server comprises a web server.

9. (Original) The system defined in Claim 1 wherein the codestream is a JPEG 2000 codestream.

10. (Currently Amended) A method comprising:

storing a compressed image as a codestream in a first progression order;

converting the codestream from the first progression order to a second progression

order different than the first progression order by reading one or more markers

of the codestream to determine a current type of progression, the one or more

markers further indicating how data of the codestream should be handled

during the progression order conversion, updating the one or more markers to

specify a target type of progression, and outputting packets of the codestream

in an order conforming to the second progression order indicated by the

updated one or more markers,

wherein the parser converts the codestream from the first progression order to an

intermediate progression order and from the intermediate progression order to

the second progression order, and

wherein the intermediate progression order is a layer progression order and the second

progression order is a target progression order other than the layer progression

order.

11. (Cancelled)

12. (Currently Amended) The method defined in Claim ~~11~~10 wherein the intermediate progression order comprises a Layer-resolution-component-position progression of JPEG 2000.

13. (Original) The method defined in Claim 12 when the first progression order is one of the following group of JPEG 2000 progression orders: resolution-layer-component-position progression; resolution-position-component-layer progression; component-position-resolution-layer progression; and position-component-resolution-layer progression.

14. (Original) The method defined in Claim 10 further comprising:  
determining where packets exist in the codestream based on at least one marker;  
creating a structure specifying components in each packet; and  
reordering packets in the codestream using the structure to map the first progression order to the second progression order.

15. (Previously Presented) The method defined in Claim 10 wherein converting the codestream from a first progression order to the second progression order is performed in response to receiving a request.

16. (Previously Presented) The method defined in Claim 15 wherein storing the codestream occurs in a memory that is part of a server that serves the image in response to requests, wherein the request is received in response to an activation by a client on a first image having the first progression order, and wherein in response to the request, a second image having the second progression order is presented to the client.

17. (Original) The method defined in Claim 16 wherein the server comprises a web server.

18. (Original) The method defined in Claim 10 wherein the codestream is a JPEG 2000 codestream.

19. (Currently Amended) An article of manufacture comprising at least one recordable media storing executable instructions thereon which, when executed by a processing device, cause the processing device to:

store a compressed image as a codestream in a first progression order; and  
convert the codestream from the first progression order to a second progression order different than the first progression order by reading one or more markers of the codestream to determine a current type of progression, the one or more markers further indicating how data of the codestream should be handled during the progression order conversion, updating the one or more markers to specify a target type of progression, and outputting packets of the codestream in an order conforming to the second progression order indicated by the updated one or more markers,

wherein the parser converts the codestream from the first progression order to an intermediate progression order and from the intermediate progression order to the second progression order, and

wherein the intermediate progression order is a layer progression order and the second progression order is a target progression order other than the layer progression order.

20. (Currently Amended) An apparatus comprising:

means for storing a compressed image as a codestream in a first progression order; and  
means for converting the codestream from the first progression order to a second progression order different than the first progression order by reading one or more markers of the codestream to determine a current type of progression, the one or more markers further indicating how data of the codestream should be handled during the progression order conversion, updating the one or more markers to specify a target type of progression, and outputting packets of the codestream in an order conforming to the second progression order indicated by the updated one or more markers,

wherein the parser converts the codestream from the first progression order to an intermediate progression order and from the intermediate progression order to the second progression order, and

wherein the intermediate progression order is a layer progression order and the second progression order is a target progression order other than the layer progression order.

21. (Previously Presented) The system defined in Claim 6, wherein the request includes a command specifying a target progression order as the second progression order.

22. (Cancelled)

23. (Previously Presented) The system defined in Claim 5, wherein the marker indicates a starting point and an ending point of data associated with the respective packet.

24. (Cancelled)

25. (Previously Presented) The system defined in Claim 23, wherein the marker indicates at least one of whether the data is to be deleted, truncated, and one or more additional operations that are to be performed on the data.

26. (Previously Presented) The system defined in Claim 23, wherein the handling information is based on rate distortion information provided via one of a PLT/PPM and a PPT/PPM marker sets.

27. (Previously Presented) The system defined in Claim 1, wherein the progression order conversion is performed using an array of packet structures, each of the packet structures corresponding to each layer of each tile in the codestream, and wherein the conversion is performed based on at least one of layer, resolution, component, and precinct progression information of the packet structures without having to decode and re-encode the codestream.

28. (Previously Presented) The method defined in Claim 15, wherein the request includes a command specifying a target progression order as the second progression order.

29. (Cancelled)

30. (Previously Presented) The method defined in Claim 14, further comprising specifying using the marker a starting point and an ending point of data associated with the respective packet.

31. (Cancelled)

32. (Previously Presented) The method defined in Claim 30, further comprising specifying using the marker at least one of whether the data is to be deleted, truncated, and one or more additional operations should be performed on the data.

33. (Previously Presented) The method defined in Claim 30, further comprising obtaining the handling information based on rate distortion information provided via one of a PLT/PPM and a PPT/PPM marker sets.

34. (Previously Presented) The method defined in Claim 10, wherein converting the codestream is performed using an array of packet structures, each of the packet structures corresponding to each layer of each tile in the codestream, and is based on at least one of layer, resolution, component, and precinct progression information of the packet structures without having to decode and re-encode the codestream.